2У

УДК 004.89

Т.И. Усова
Одесский национальный политехнический университет, г. Одесса, Украина
usovati@rambler.ru

Использование экспертной системы в информационной технологии распараллеливания решения нелинейных уравнений

Рассматривается проблема оптимизации работы автоматической системы распараллеливания решения нелинейных уравнений с целью дальнейшей реализации на вычислительном кластере. Предлагается методика построения информационного графа нелинейного уравнения на основании элементарных структурных единиц с использованием экспертной системы.

Введение

Решение нелинейных задач является одной из главных проблем в любой области науки и техники. Методы, используемые на данный момент для их решения, зачастую не удовлетворяют критерию быстродействия; для ускорения работы этих методов следует их распараллеливать. С учетом существующей проблемы автоматического построения информационного графа (ИГ) решения уравнения, предложено использовать экспертную систему.

Целью данной работы является автоматизация распараллеливания решения нелинейных уравнений. Предлагаемая система программного распараллеливания решения нелинейных уравнений предназначена для автоматизации процесса преобразования алгоритма нахождения корней сложных алгебраических и трансцендентных уравнений в параллельную программу.

Задача заключается в разработке методики построения ИГ решения уравнения с использованием экспертной системы.

Алгоритм представления решения уравнения в виде структурно-процедурной программы

В последнее время получили широкое применение многопроцессорные вычислительные системы (МВС), что дает возможность развития алгоритмов распараллеливания решения сложных нелинейных уравнений (НУ), которые часто встречаются в прикладных науках. Проблема заключается в том, что для каждой такой задачи часто необходимо разрабатывать свой собственный метод решения, так как большинство видов НУ [1] не имеют аналитического решения.

Алгоритмическое обеспечение параллельных вычислительных систем в приложении к конкретным вычислительным задачам является на данный момент одним из наиболее актуальных направлений исследований. Так как создание эффективной параллельной программы для МВС не всегда бывает успешным.

Для представления решения любого нелинейного уравнения используется модель в виде информационного графа [1].

Информационный граф (ИГ) – граф алгоритма решения задачи – строится на основе разбиения исходного уравнения и алгоритма решения уравнения подобного типа на элементарные структурные единицы [1]. Данный этап структурного анализа решаемого нелинейного уравнения позволяет не задумываться о его виде и сложности решения, так как все уравнения состоят из элементарных операций в разных комбинациях (рис. 1). Полученные структурные единицы представляют собой подграфы ИГ. В зависимости от сложности первоначальной структуры уравнения разбиение может происходить или в один этап, или в несколько – рекурсивно, до тех пор, пока полученные элементы или не будут содержать знаки операций или числа, или сами не будут представлять те виды простейших структурных единиц, которые были приняты за элементарные.

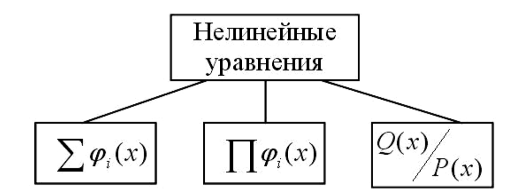


Рисунок 1 – Разбиение нелинейных уравнений по структуре

Была разработана информационная технология распараллеливания нелинейных уравнений. Она состоит из следующих этапов:

1. Выделение элементарных структур. На входе системы мы получаем сложное алгебраическое или трансцендентное уравнение. На выходе генерируется совокупность структурных единиц, из которых оно состоит. На данном этапе процесс выделения структур может координироваться экспертом.

2. Полученная на первом этапе совокупность структурных единиц представляется в виде информационного графа. В качестве узлов графа могут выступать только числа и знаки операций. Если на этапе разбиения были использованы элементарные структурные единицы, определенные нами, то здесь они представляются уже не одним узлом, а подграфом.

3. Строится информационный граф алгоритма метода решения нелинейного уравнения, определяемого экспертом. Построенные на этапе 2 информационные графы уравнения и его производных объединяются с основным графом – графом алгоритма нахождения корней уравнения.

4. Преобразование информационного графа решения уравнения в кадровую ярусно-параллельную форму.

5. Ярусно-параллельная форма информационного графа преобразуется в структурно-процедурную форму с учетом того, что кадры располагаются по уровням, а каждый кадр рассматривается в качестве фундаментального оператора [2]. В конце этого этапа создается структурно-процедурная программа (алгоритм представления решения уравнения в виде структурно-процедурной программы представлен на рис. 2), которая может быть реализована на многопроцессорной системе.

Исходными данными для получения структурно-процедурной программы решения нелинейного уравнения являются:

1. U – сложное алгебраическое или трансцендентное уравнение;
2. a,b – границы отрезка локализации корня;
3. ε – точность решения уравнения.

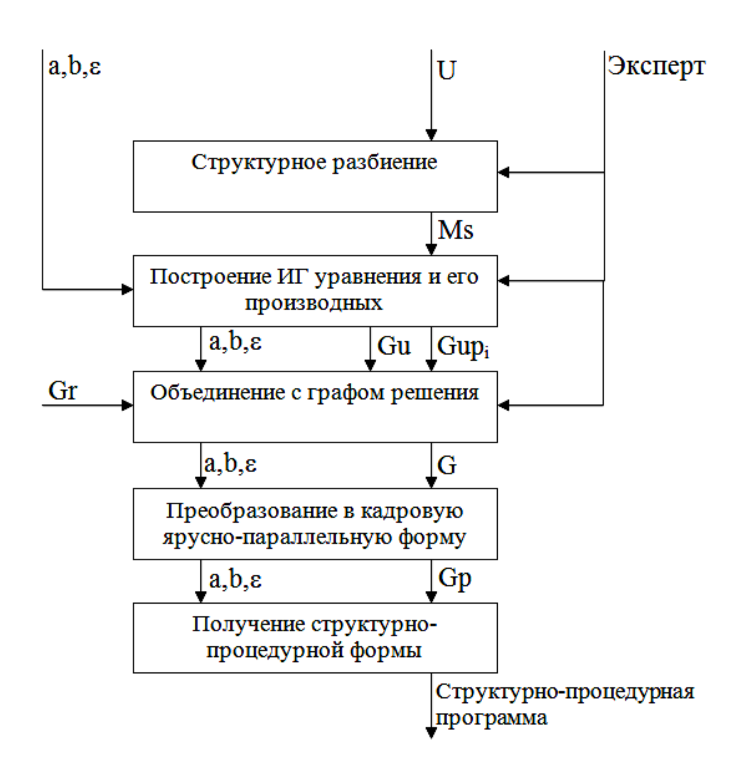


Рисунок 2 – Алгоритм представления решения уравнения
в виде структурно-процедурной программы

Во время работы экспертной системы используется база знаний (БЗ), в которой хранится совокупность элементарных структурных единиц уравнений (Ms) и некоторые алгоритмы их решения, уже представленные в виде графов (Gr).

В процессе работы экспертной системы происходит построение информационного графа уравнения (Gu) и, в зависимости от выбранного алгоритма решения, информационных графов производных (Gup_i – информационный граф i-й производной). На основании построенных графов уравнения, производных и алгоритма нахождения корней строится общий граф решения (G).

После окончания работы экспертной системы полученный граф решения автоматически преобразуется в кадровую ярусно-параллельную форму (Gp).

Распараллеливание данного решения получило развитие с использованием структурно-процедурной реализации вычислений для систем с массовым параллелизмом [3], когда исходный ИГ преобразовывается в кадровую форму, где каждый кадр представляет собой программно-неделимый участок кода (структурная компонента), а набор кадров является процедурной компонентой, последовательно выполняющейся на МВС. Каждый кадр внутри может содержать множество операционных вершин, которые являются независимыми друг от друга и выполняются параллельно. С учетом того, что каждый кадр может содержать произвольное число операционных вершин (от одной до нескольких сотен для сложных задач), неизвестно, сколько именно процессоров будет задействовано.

Решение такой проблемы предложено в [4], где ИГ представляется в ярусно-конвейерной форме, для более четкого определения предшествующих и последующих вершин. Каждой вершине ставится в соответствие свой процессор с учетом проложения к нему наименьшей трассы с использованием карты волновых фронтов для определения возможности участия данного процессора в формировании маршрута. Если для каких-то узлов МВС трассу создать не удается, происходит откат и новое формирование трассы на основе других узлов. Этот алгоритм может быть использован для различных коммутационных структур и для графов, содержащих как обратные связи, так и цикли-

ческие участки. Однако его использование целесообразно для масштабных задач, когда время трассировки будет несоизмеримо мало по сравнению со временем вычислений. В нашем случае ситуация иная, и время, которое будет потрачено на трассировку, значительно ухудшит временные показатели вычислений самого НУ. В связи с этим предлагается использовать динамические структуры для задания количества узлов кластера, предполагая, что время пересылки данных между ними одинаково. В этом случае можно использовать структурно-процедурную реализацию вычислений из [3] для кластера, когда процедурная компонента, переходя от кадра к кадру, будет динамически изменять число используемых процессоров, что приведет к более разумному использованию ресурсов.

Роль экспертной системы при построении ИГ

На этапе разбиения нелинейного уравнения по структуре используется база знаний, содержащая в себе набор уже известных элементарных структурных единиц. В процедурах структурного разбиения и построения информационных графов принимает участие эксперт, имея возможность внести свои коррективы в работу системы.

Нелинейное уравнение должно быть представлено в виде графа определенного вида, так как в дальнейшем он будет объединяться с информационным графом алгоритма решения уравнения данного типа. В зависимости от вида решаемого уравнения алгоритм его решения может отличаться, но, учитывая, что большинство нелинейных уравнений не имеет аналитического (точного) решения, система автоматически выбирает один из алгоритмов итерационного нахождения корней. В этом случае часто приходится множество раз вычислять значение функции при различных значениях аргумента. Кроме того, большинство итерационных методов решения использует производные первого порядка и выше. Соответственно, необходимо получение аналитического вида этих производных, а также информационных графов, соответствующих им. Если получение аналитического выражения для производных невозможно, система может использовать алгоритмы численных методов нахождения производных функций, хранящиеся в ее БЗ. Именно от эксперта будет зависеть, какой алгоритм решения и какая точность вычисления производных (если в этом есть необходимость) будут выбраны для заданного уравнения. Далее в табл. 1 приводится методика построения ИГ уравнения, не имеющего аналитического решения, с использованием экспертной системы и уточнением роли эксперта на каждом шаге.

Таблица 1 – Роль эксперта в методике построения ИГ

Шаг построения ИГ	Роль эксперта
1. Синтаксический анализ уравнения	Преобразование уравнения к определенному виду (если необходимо)
2. Построение синтаксического дерева	
3. Распознание в дереве операций структурных единиц из базы знаний	Возможность добавления новых структурных единиц в БЗ, при отсутствии необходимых
4. Построение дерева вычисления уравнения с учетом выявленных структур	
5. Подстановка констант в узлы дерева и определение входных данных	
6. Получение ИГ вычисления уравнения (Gu) на основе дерева вычисления	Контроль количества операций на каждом уровне
7. Выбор алгоритма решения уравнения и его графа (Gr) из БЗ	Назначение системе алгоритма решения, в зависимости от выбранного алгоритма, назначение построения дополнительных графов
8. Построение ИГ решения	Окончательное решение об оптимальности ИГ

Конечно, вполне вероятно возникновение такой ситуации, когда система не сможет найти в БЗ структурной единицы, соответствующей заданному уравнению. В этом случае эксперт может пополнить набор за счет добавления новых структур в процессе эксплуатации системы или, если это будет возможно, собрать необходимую структуру из имеющихся в базе элементов, что не всегда удобно, так как может увеличить количество выполняемых операций при вычислении уравнения. Добавление новых структурных единиц осуществляется при участии эксперта с помощью редактора базы знаний. Новые элементы в этом случае могут быть созданы на основе уже существующих.

В качестве примера работы экспертной системы рассмотрим построение ИГ для решения уравнения

$$(\sin(\lg x)+\lg x)(\cos x+(x-5)(x+3))=0. \qquad (1)$$

На первом шаге происходит автоматический синтаксический анализ исходного уравнения и строится синтаксическое дерево (рис. 3).

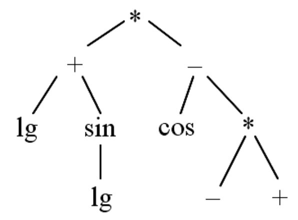


Рисунок 3 – Синтаксическое дерево уравнения

Далее происходит распознание элементарных структурных единиц уравнения из тех, что хранятся в базе знаний. В нашем случае могут быть использованы структурные единицы, представленные на рис. 4.

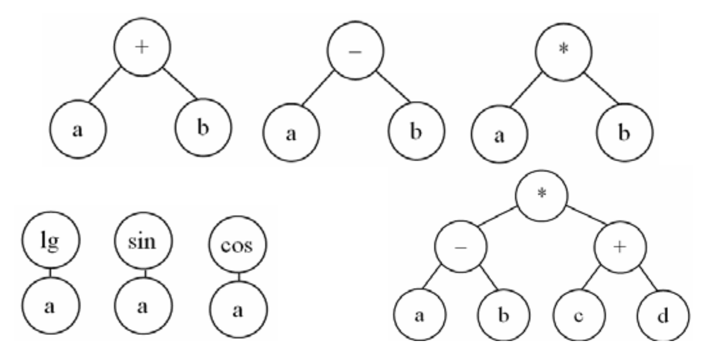


Рисунок 4 – Некоторые элементарные структурные единицы уравнения

После построения дерева вычисления уравнения с учетом выявленных структур, подстановки констант в узлы дерева и определения входных данных будет получен ИГ вычисления уравнения (Gu) (рис. 5). Здесь мы видим некоторые недочеты при построении графа. Во-первых, здесь два раза повторяется операция логарифмирования над одним и тем же аргументом. Количество выполняемых операций всегда желательно уменьшать. Во-вторых, здесь на первом уровне должно выполняться четыре операции, а на втором – три. В случае исправления ситуации с двойным вычислением логарифма, на первом уровне остается четыре операции, а на втором – всего две. Эту ситуацию можно исправить переносом операции косинуса на второй уровень. Подобные недочеты кажутся вполне безобидными в случае, если вычисления будут происходить только один раз, но когда этот процесс будет происходить в цикле, это может отразиться на скорости вычислений. Кроме того, так как в дальнейшем на основании этого графа будет получена структурно-процедурная программа, предназначенная для выполнения многопро-

цессорной системой, желательно, чтобы нагрузка на узлы системы была как можно более сбалансированной. Таким образом, ИГ решения уравнения, который представлен на рис. 6, более соответствует поставленной задаче.

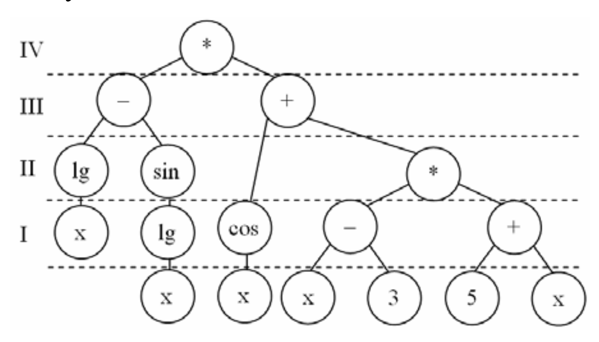


Рисунок 5 – ИГ уравнения до коррекции экспертом

После того, как был построен граф вычисления уравнения и выбран алгоритм нахождения корней, решается вопрос о нахождении производных. В нашем случае для нахождения корней был выбран комбинированный метод хорд-касательных [1], а для вычисления значений производных построены информационные графы на основе аналитических выражений. Теперь, когда собраны все составляющие, может быть построен полный ИГ решения уравнения (рис. 7) для получения структурно-процедурной программы.

Достаточно сложным представляется хранение в базе знаний структурных единиц. Наиболее подходящей для этих целей является сетевая модель.

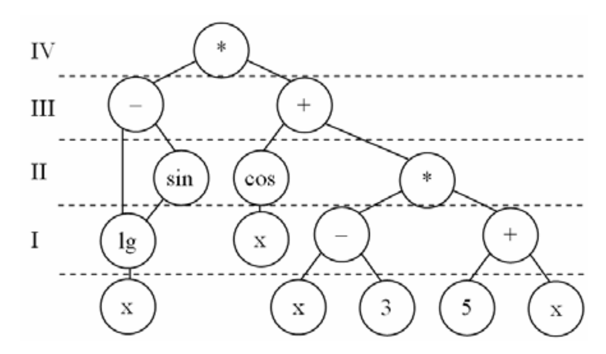


Рисунок 6 – ИГ уравнения после коррекции экспертом

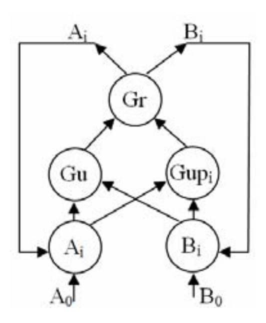


Рисунок 7 – ИГ решения уравнения

Выбор сетевой модели знаний обусловлен тем, что в ее основе лежит идея о том, что любые знания можно представить в виде совокупности объектов (понятий) и связей (отношений) между ними. В отличие от продукционных эти модели более наглядны, поскольку любой пример можно представить в виде ориентированного (направленного) графа, что, собственно, и является нашей задачей.

Сетевая модель данных – логическая модель данных, являющаяся расширением иерархического подхода. В отличие от иерархической, в сетевой модели данных у потомка может быть любое число предков, а в иерархических структурах запись-потомок должна иметь только одного предка.

Хранимые в БЗ структурные единицы изначально должны представлять собой графы. Операции с хранимыми структурными единицами соответствуют основным операциям манипулирования данными в сетевой модели данных и помогают осуществить построение ИГ с наименьшими усилиями, к ним относится:

1) нахождение конкретной записи в наборе;
2) поиск более сложной (или простой) структуры переходя к следующему потомку или предку по некоторой связи;
3) создание новой структуры, в случае отсутствия необходимой;
4) изменение существующей структуры до полного соответствия поставленной задаче;
5) включение в связь или исключение из связи для данного построения.

Перечисленных операций достаточно для построения информационного графа решения уравнения практически любой сложности.

Выводы

Разработанная методика автоматизации построения ИГ с использованием экспертной системы позволяет распараллеливать различные виды задач из класса нелинейных за счет инвариантности модели решений относительно вида уравнений и дает возможность объединить алгоритм структурного анализа нелинейного уравнения и структурно-процедурную реализацию на многопроцессорной системе с программируемой архитектурой.

Литература

1. Паулин О.Н. Метод распараллеливания нелинейных задач / О.Н. Паулин, Т.И. Усова // Проблемы программирования. – 2010. – № 2-3. – С. 145-148.
2. Реконфигурируемые мультиконвейерные вычислительные структуры // [Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.]. – Ростов-на-Дону : ЮНЦ РАН. – 2008. – 320 с.
3. Левин И.И. Ресурсонезависимое параллельное программирование / И.И. Левин // Материалы Международной научно-технической конференции «Искусственный интеллект» – 2002. – Таганрог : ТРТУ, 2002. – С. 194-198.
4. Левин И.И. Реализация информационных графов в архитектуре многопроцессорных систем / И.И. Левин // Материалы Международной научно-технической конференции «СуперЭВМ и многопроцессорные вычислительные системы» – 2002. – Таганрог : ТРТУ, 2002. – С. 98-102.

Т.І. Усова
Використання експертної системи в інформаційній технології розпаралелювання розв'язання нелінійних рівнянь
Розглядається проблема оптимізації роботи автоматичної системи розпаралелювання розв'язку нелінійних рівнянь з метою подальшої реалізації на обчислювальному кластері. Пропонується методика побудови інформаційного графа нелінійного рівняння на основі елементарних структурних одиниць з використанням експертної системи.

T. Usova
The Expert System Application in the Information Technology for Parallel Solving of Nonlinear Equations
Problem of automatic system for parallel solving of nonlinear equations performance optimization is considered for the purpose of further implementation on the computing cluster. The information graph of nonlinear equation construction methods based on elementary base units with expert system application is represented.

Статья поступила в редакцию 01.07.2010.